Advances in Computer Science and Information Technology (ACSIT)
p-ISSN: 2393-9907; e-ISSN: 2393-9915; Volume 3, Issue 1; January-March, 2016, pp. 22-26

http://www.krishisanskriti.org/Publication.html

Educational Game with New Additional Function

Km. Archana Patel

Student of Computer Application NIT KURUKHSETRA (Haryana)
E-mail: archanamca92@gmail.com

Abstract—*Scratch editor is visual programming atmosphere that enables users to learn computer programming while functioning on personally significant project likes games and animated stories. Scratch is designed to remove initial difficulties of the programmer who is beginner . A key goal of Scratch is to introduce programming to those who do not have preceding programming skill. The Scratch application is used to generate projects containing scripts and media. However in scratch editor several functions are not offered that is why the programmer has to develop code himself due to this reason programmer code become extremely large. In this paper firstly i have developed an educational game then through this educational game i have shown how to increase the functionality of scratch editor by creating own block which is not available in scratch editor. After adding my own block, scratch editor worked effectively and this is very convenient for user because my code length is very less comparative to before and it takes very less time.*

Keyword*: Fetch drop Programming Language, Game, Scratch Editor and block, Active Programming, Tinkerability.*

1. INTRODUCTION

It provides a full set of multimedia tools you can use to generate fantastic applications, and you can do so without any difficulty as compared to other programming languages. People make animations, stories or games[6]. Scratch was planned with the goal of creating it simple for anyone, mainly youth, to learn the program. Not simply is knowledge of programming important in the digital age, however Scratch conjointly supports systems thinking and computational[17] and complexity solving in different domains . Scratch can offer a atmosphere during which lecturers and students both are learning by designing. While without programming information[7], users easily accesses scratch editor. This goal drove several aspects of the Scratch style. Some of the design selection are , like the selection of a visual blocks language the minimal command set and also the single-window user interface layout. Others are less noticeable, like however the target viewers influenced the sort system and also the approach to error handling. This text explores aspects of the Scratch programming atmosphere and language style that build it easier for people to express themselves, explore and learn. Rather than typing commands, programming in Scratch is performed by combination of programming blocks[8,1] and dragging. This graphical interface permits users to simply manage the way during which different kind of commands react to each other. Moreover, every block can set with another on condition that it makes computation sense. The most effective aspect of fetch-drop programming languages is providing students visualized code structures and conjointly their relationship[3]. These structures' association may be the foremost difficult part of the programming. Scratch is programming rule for everybody. It helps to form interactive art, music, games and stories and share them on-line[4]. The author of this study is also concerned during this study as a teacher and as a student. Result of this study shows that exploitation Scratch makes programming additional pleasurable, helps learning algorithmic concept, and additional visual.

The remaining part of this paper is organised as follows. Section 2 gives study about scratch environment. In Section 3 we have defined programming language. In section 4 we talks about our proposed work. Section 5 gives a result about functionality. Section 6 presents a conclusion and future work.

2. ENVIRONMENT

Scratch is such associate that gives opportunities for the people to form their own prototypes without knowing high level of programming rules[5]. The visual programming technique in Scratch helps people to get meanings and to make connections. The commands are divided into eight classes like control, Sound, looks and motion. This avoids long list of commands, doubtless overwhelming.

A key feature of Scratch is that it continuously works. There is no edit/run mode distinction or no compilation step. Users will click on a command or program fragment at any time to check what it does. In fact, they can even add blocks to a script or modify parameter while it is running. By eliminating compilation pauses and doubtless jarring mode switches, Scratch helps users keep engaged in testing, debugging and rising their projects . I mean to say that Scratch is a tinkerable. Tinkerability encourages active learning and supports a bottom-up approach to writing scripts wherever tiny chunks of code are tested and assembled, then combined into bigger units. Tinkerability helps programmer to determine the functionality of blocks [19]. A block is often tested by clicking on it, even within the palette. Every block comes with default parameters that offer associate illuminating demonstration of

what that block does. Scratch has facilitate screens for each command, accessible via the right-button menu, however several users study commands simply by attempting them. Scratch does not need it from user to produce complete scripts before running the project. Program fragments can often left in the scripting pane and are saved with the project.

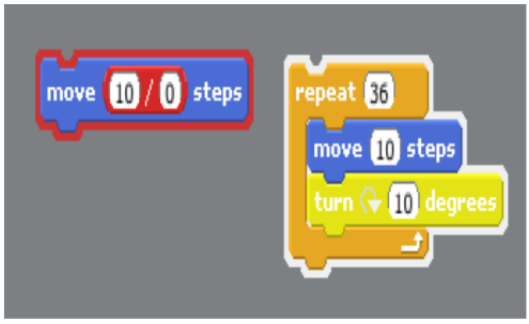

Fig. 1: Feedback for an error is red

Such fragments play a role like commented-out code in an text-based language. Once troubleshooting, an extended script are often broken into piece and every piece tested independently[19]. Scratch gives visual feedback to indicate script execution. Once a script is running, it is enclosed by a glowing white border. This feedback helps the user to recognize when scripts are measure triggered and how long they run. If a script encounters an error such as dividing by zero, the border or corner turns red and therefore the block that caused the error is decorated in red (Figure 1). Similarly, Scratch In most text-based programming rule, variables are abstract, invisible. Syntax errors are removed because, blocks fit together only in ways that makes sense, Scratch turns variables into concrete objects that the user can see and manipulate, creating them easier to know through observation and tinkering.

3. PROGRAMMING LANGUAGE

Scratch scripts are engineered by snapping jointly blocks representing statements, control structure and expression[8]. The design of the blocks propose how they match and fit together, and also the fetch-and-drop system refuses to attach blocks in ways that would be insignificant. In Scratch, the visual grammer of block design and their combination rules play the role of syntax during a text-based language. There are four varieties of Scratch blocks: function blocks, command blocks, control structure blocks and trigger blocks. When command blocks are snapped along to make a sequence of commands, or stack, the notches and bumps match along like puzzle items. Control structure blocks are type of command block with one or additional nested command sequences. The shape of control structure blocks creates them simple to use. In Scratch, control structure block is an indivisible unit. Instructors using Scratch as a fast introduction to programming before change to a text-based language report that some people continue to 'think in Scratch blocks' as a variety of pseudo_code, even after moving to the text-based rule. Command blocks are just like the statements of a text-based rule; function blocks are like operators. Function blocks are not gathered in linear sequences like command blocks. As an alternative, they are used as arguments to commands and nested along to make expressions. Once assembling scripts, Scratch solely permits blocks to be connected in meaningful ways. A command block connects when dropped into command sequence, however a function block will not connect if dropped in the same place. Because the user drags a block, Scratch provides visual feedback [14,20] showing parameter slot targets (function blocks) or attainable sequence insertion points (command blocks)

4. PROPOSED WORK

Every research result shows that visual programming can be more efficient than the classical textual programming, students can be more aggravated, less fed up and not overloaded with the syntax of programming language. Scratch can provide an environment in which teachers and students both are learning by designing. Users have to write very big code for program because in scratch editor many function such as Jump(), run(), go to(), call() are not available. Therefore users faced many problem when they works in scratch editor. Firstly I shall describe our game then by help of game I have shown that our code is very large due to absent of remaining function which is not available in scratch.

Our game named **Catching Alphabets** for children of period between 2 - 5year to teach them alphabets and number or many things through game.The Idea of this game is that children will use a object to move over cloud to cloud and it has to cover maximum obstacle(catching the alphabet) those coming in its path. More obstacle in minimum time will help children to make more score, with each obstacle we attached sound(name of alphabets) of that type of obstacle so when children will play this game they could learn alphabets and number with fun.

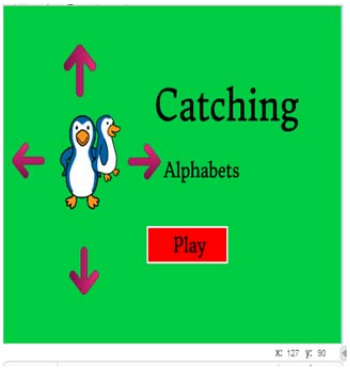

Fig. 2: Four objects for each direction

That's easy way to teach children alphabets and number with highly entertaining them. So we took an object(Penguin) which will be handle by children, here it is

Fig. 3: Object

In code of my game i took gravity to handle speed of velocity and velocity for drop down of this object until it reaches to zero, so when we press up key it will hike the value of gravity to 11. Gravity and velocity will took it down slowly until it reach to zero or it not touches cloud surface.

Fig. 4: Object with velocity and gravity

During this time lots of obstacle like number and alphabets will come in its way so it has to cover them to listen corresponding sound of that obstacle.

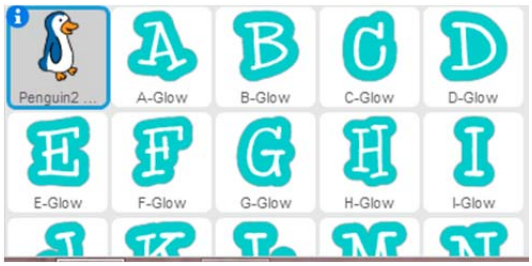

Fig. 5: Obstacles corresponding sound

In this figure we use all alphabets as obstacle and we attached corresponding sound with each and every alphabets. When object will touch to any one of these obstacle, it will play sound related to that obstacle. We also used various costumes of that object that will change when it will jump.

Fig. 6: Four costumes

If we add these kind of block than it will also helpful for non programmers users to make application more effective

Fig. 7: All variables

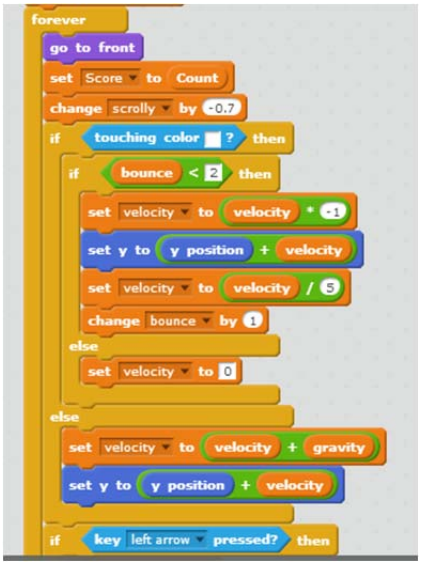

Fig. 8: Value corresponding to variables

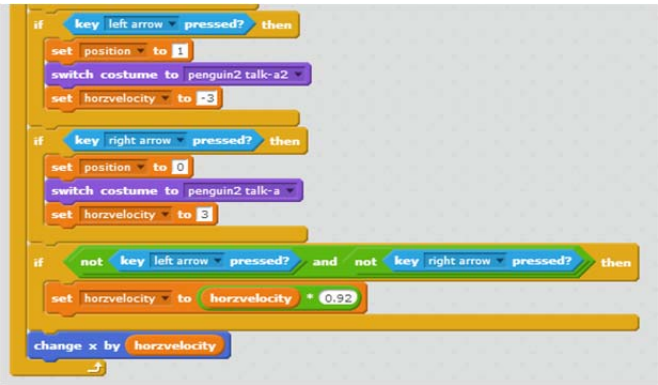

Fig. 9: Objects set to left and right direction

Every time value of velocity and gravity decreasing because we are dividing it by 5. Which take it near to zero after few loop instruction Coding part of jump is

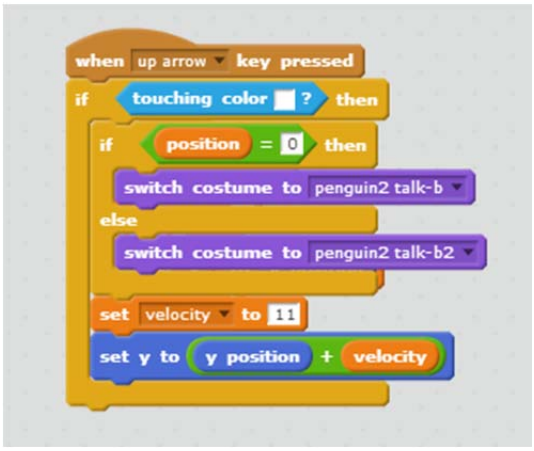

Fig. 10: Object set up and down direction

Here we increasing value of velocity with 11, which helps object to move up side and gravity uses against this. so it will go down after 11 steps. We made layer scroll and score function to make more perfect.

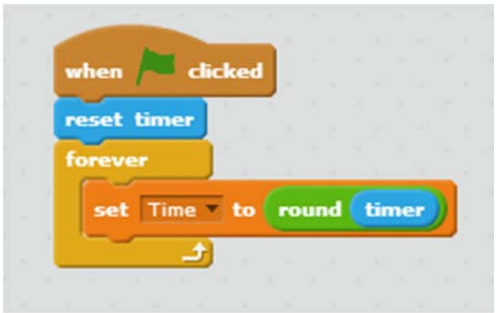

Fig. 11: Set the value of timer

In score function we are taking round figure of time which are in mille second so we are dividing it by 10. Here I am increasing value of velocity with 11, which helps object to move up side and gravity uses against this. so it will go down after 11 steps. I made layer scroll and score function to make more perfect.

Fig. 12: Score value

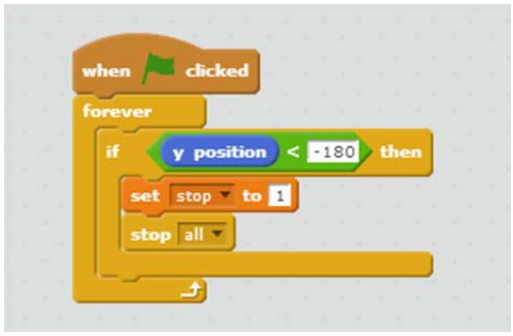

Fig. 13: Stopping range

Here score is a variable whose value started with zero and here 360*0 shows 1 layer y co-ordinates value where this sprite going to set in real. For next layer we will use 360*1 and for further 360*2 and so on

In this program our code is too long because we used jump and scoring function which is not available in scratch editor. So our main focus is , put this code jump and scoring code into one single block and given the name of this block is Jump so that our program length became very less comparative to previous code .Jump code used for two purpose one for when our sprite falldown and another for stopping the sprite Scoring block gives the total score of the program. We put these code into corresponding one single block. Suppose we put this block into three different block and given that name of corresponding block is jump down , jump up and scoring block. Now our full code is into the only single block.

Fig. 14: Our own block

So if users wants to write a code for jump and score function then he has to fetch the jump block and scoring block then drop it in to the code area. If i use these single blocks then our big code converted in to small code because these big blocks contain all small block into one single block. So if users use

these blocks then he will take very less time for writing a code and it is more efficient comparatively to write long code.

5. RESULT

After completing my game I found that if i want to write code without using jump function and scoring function then my code length is very long and its take very long time also it is very inconvenient . But after making my block, i fetched block only and do not need to write a long code. I am using only three single blocks instead of long code. So our scratch editor worked effectively and this is very convenient for users because my code length is very less and it takes less time.

6. FUTURE SCOPE & CONCLUSION

In coding of my game I have mainly focused on two things one is easy programming for the users and other is less time consuming. For these two things I have added functions which are not available in scratch.

If my own block added into the scratch editor then it will work effectively.

In future this can be more beneficial for educational as well many fields related with scratch editor by adding up many relevant features which will be develop by further research.

REFERENCE

[1] Stamatios Papadakis, Michail Kalogiannakis, Vasileios Orfanakis, Nicholas Zaranis: Novice Programming Environments. Scratch & App Inventor: a first comparison, *IDEE'14*, June 9, 2014

[2] Quinn Burke, Yasmin B. Kafai: The Writers' Workshop for Youth Programmers Digital Storytelling with Scratch in Middle School Classrooms,February 29–March 3, 2012,

[3] Brett Ward, Tim Bell, Daniela Marghitu, Lynn Lambert: Teaching computer science concepts in scratch and alice: 2010

[4] Amanda M. Bell: Learning Complex Systems with Story-Building in Scratch: IDC 2015 Medford, MA, USA

[5] Nurul Syuhada Joini, Nurfauza Jali, Syahrul Nizam Junaini: An Interactive Mathematics Game Using Scratch Programming: research gate 2015

[6] Mehmetcan Fal, Nergiz Ercil Cagiltay: how scratch programmıng may enrıch engıneerıng educatıon : 2nd International Engineering Education Conference:2013

[7] Ahmet Baytak , Susan M. Land: An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom, 7 February 2011.

[8] Jeffrey V. Nickerson, Andrés Monroy-Hernández: Appropriation and Creativity: User-Initiated Contests in Scratch, Hawaii International Conference on System Sciences – 2011

[9] Thierry Nodenot. Pierre Caron, Xavier Le Pallec, Pierre Laforcade:Applying Model Driven Engineering Techniquesand Tools to the Planets Game Learning Scenario: journal ofinteractive media in education 2008

[10] Chih-Kai Chang: Effects of Using Alice and Scratch in an Introductory Programming Course for Corrective Instruction: journal of educational computing research: 2014

[11] Sotamaa, O: Play, Create, Share? Console Gaming, Player Production and Agency: The Fibreculture Journal 16, 2010.

[12] Jeppesen, L. B. "Profiting from Innovative User Communities. How Firms Organize the Production of User Modifications in the Computer Games Industry,": 2015

[13] Annemarie Harzl, Philipp Neidh¨ofer, Valentin Rock, Maximilian Schafzahl, and Wolfgang Slany: A Scratch-like visual programming system for Microsoft Windows Phone 8: 6 oct 2013

[14] Ken-Yu Lin, JeongWook Son, Eddy M. Rojas: A Pilot study of a 3d game environment for construction safety education: journal of Information Technology in Construction - ISSN 1874-4753: January 2011

[15] Kafai, Y. B. & Peppler, K. A: Developing Gaming Fluencies with Scratch: Realizing Game Design as an Artistic Process: research gate(Jan 2012)

[16] Jui-Feng Weng, Shian-Shyong Tseng: Teaching Boolean Logic through Game Rule Tuning: IEEE transactions on learning technologies, vol. 3, no. 4, october-december 2010

[17] Matthew, john ventura, chad baker: Development of a Video Game that Teaches the Fundamentals of Computer Programming: Proceedings of the IEEE SoutheastCon 2015, April 9 - 12, 2015

[18] Shute, V. J. & Ventura, M. (2013). Stealth Assessment: Measuring and Supporting Learning in Video Games. Cambridge, MA: The MIT press.

[19] John maloney, Mitchel resnick, Natalie rusk, Brian silverman, and Evelyn eastmond: The scratch programming language And environmentacm transactions on computing education, vol. 10, no. 4, article 16, pub. Date: november 2010.

[20] Florian Mehm : Authoring Serious Games :FDG 2010, June 19-21, Monterey, CA, USA

[21] Lindsay grace, peter jamieson, naoki mizuno: VerilogTown: Cars, Crashes and Hardware Design: ACE 2015

[22] Blikstein, P. 2013. Gears of our childhood: constructionist toolkits, robotics, and physical computing, past and future. In Proceedings of the 12th International Conference on Interaction Design and Children (pp. 173-182). ACM.

[23] Langdon, D., McKittrick, G., Beede, D., Khan, B., & Doms, M. 2011. STEM: Good Jobs Now and for the Future. ESA Issue Brief# 03-11. US Department of Commerce

[24] Quinn, A. J., & Bederson, B. B. 2011. Human computation: a survey and taxonomy of a growing field. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 1403-1412). ACM.